(2)

AD-A231 543

A RAND NOTE



# A Conceptual Design for the Model Integration and Management System

Bart Bennett

April 1989





RAND

This publication was supported by The RAND Corporation as part of its program of public service.



Published by The RAND Corporation
1700 Main Street, P.O. Box 2138, Santa Monica, CA 90406-2138

A RAND NOTE

N-2645-RC

# A Conceptual Design for the Model Integration and Management System

Bart Bennett

April 1989

MOSF
MILITARY OPERATIONS SIMULATION FACILITY
THE RAND CORPORATION

RAND

## PREFACE

The Model Integration and Management System (MIMS) is an outgrowth of the Military Operations Simulation Facility (MOSF) and the overall frustration incurred while trying to apply models as analytic decision and policy analysis aids when corresponding projects have time, budget, or personnel limitations. Researchers have been searching for "a better way" to rapidly integrate internally developed or externally obtained models into an environment that includes a robust set of modeling and analysis tools. Typically, these models are required only at specific times and only for short periods. Hence, a system is needed that reduces model maintainability requirements and, at the same time, allows for rapid reusability. Prior approaches have led to single model solutions that are satisfying only in the very short term. A broader, more comprehensive solution scheme has been developed here which relies on functional components and automated, transparent bridges between the analyst's perspective and the model's environment.

The purpose of this Note is to provide the motivation for developing an enhanced modeling environment and to describe the conceptual architecture for the MIMS. In particular, emphasis is placed on the analyst's perspective and the need to perform tactical and strategic military-oriented quantitative analysis in a simpler, more effective way. The methodology described herein should be of interest to analysts and research managers who rely on the successful use of models. Model designers and implementers, as well as data managers or information scientists, will also find this Note intriguing.

This work was funded solely by The RAND Corporation but has received many synergistic benefits from the MOSF experience gained in responding to the needs of Project AIR FORCE, the National Defense Research Institute, and the Arroyo Center.

## SUMMARY

The Model Integration and Management System (MIMS) is a structured modeling framework designed to simplify the installation or creation of models within a common environment as part of RAND's Military Operations Simulation Facility (MOSF) (Donohue et al., 1986). The goal of the MOSF is to enhance the quality of tactical and strategic military analysis and the effectiveness with which it is performed. Current operational requirements associated with applying a model are so restrictive that an alternative, "lower cost" system is needed. This system must allow the analyst to interface more easily with the model without having to make changes to it.

The MIMS is designed with a high degree of generality, allowing commonly used tools (e.g., graphics, data management, statistics, and mathematical algorithms) to be readily accessible. The analyst is provided with a standard set of model-independent interfaces to these tools and the models and therefore need only learn one set of "user-friendly" protocols. In addition, there is no need to alter the model, eliminating one source of extensive delays.

The MIMS uses decision support system, knowledge-based system, and intelligent database methodologies that interface with the analyst and automate many of the repetitive, tedious tasks performed by database and *modeling technical analysts* and programmers. Requisite database information including formats and data relationships are stored as templates which can be used to automatically interpret the source and model databases. Tools to perform data integration, scenario generation, and results analysis are provided for the analyst.

By no means do we intend to infer that the MIMS will solve all modeling problems. The MIMS concept is specifically aimed at reducing the tedious data-oriented aspects of modeling in the same way that high level language compilers have relieved us of the burden of programming 0's and 1's. It can be viewed as a modeling operating system that invites enhancement and extension rather than shying away into its own environment. Ideally, the MIMS is its own environment, but at the same time, it can be encapsulated within other environments. It poses no restriction on the target models and allows tailoring to suit the user.

Detailed specifications for this seemingly ideal, but difficult to create, modeling system will be described in a forthcoming system requirements document.

## ACKNOWLEDGMENTS

The Model Integration and Management System (MIMS) concept could not have been solidified without the efforts of a number of RAND colleagues. The many interactions I have had on various national security research projects motivated me to think a little loftier and to envision a more general means for manipulating models.

Special thanks to Roy Gates and Judy Lender who worked with the MIMS concept and produced a magnificent view of the system requirements. They helped to refine the initial ideas and actually made the concept come alive. Their tireless efforts to bring the dream the first step toward reality are greatly appreciated. They also provided much assistance in reviewing and helping to put this manuscript together.

Jon Hertzog and Stephen Drezner reviewed the original MIMS proposal, provided much guidance, and were instrumental in getting Phase I of the project funded. I am grateful to Donald Rice, President of The RAND Corporation, for having enough confidence in our work to support it with RAND's own funds. I appreciate their combined dedication on behalf of the MOSF in general. In addition, thought-provoking discussions with Stephanie Cammarata, Iris Kameny, and Herbert Shukiar have helped to purify my ideas. Stephanie provided a detailed review of the document that led to a variety of significant improvements. RAND's Publications Department personnel are specially acknowledged for their concern, care, and attention given to preparing the manuscript.

Finally, my deepest appreciation goes to my former graduate advisor, Dr. Eliezer Naddor, who shared this vision and always knew that this sort of thing could and must be done.

## CONTENTS

## FIGURES

# I. INTRODUCTION AND MOTIVATION

Military and policy analysts use computerized models and simulations as decision aids for evaluating, testing, and comparing tactical and strategic policies. Because of the complex issues under investigation, these models tend to be extremely intricate and large-scale, and, hence, difficult to manage or adapt. An enormous effort is required to use a typical model, much of which deals with understanding the idiosyncrasies of the associated databases and manipulating the data into the proper form. The lack of analyst-oriented support tools hinders the effective application of models in two ways. First, analysts who design models do not have a readily available, integrated support system of interfaces, graphics, data management, statistical/mathematical algorithms, and inter-model communications tools. Therefore, model builders must include this functionality in with their computational models in a nonstandard and often inconsistent way. Second, analysts, wishing to apply a particular model, must either become model and database experts or must employ the designers or other trained experts. In most models, a novice user simply cannot manipulate the highly specialized data-structuring and interactive communications required for the model. This current state of modeling poses unacceptable requirements on already scarce resources — manpower, time, and money. The continuance of these practices ultimately impedes the quality of analytical research.

## THE SAGE FAMILY OF MODELS

Currently, the Sequential Analytic Game Evaluation (SAGE) modeling methodology is a critical resource at RAND.[1] This technique, employing two-sided zero-sum game theory, was first used in the Tactical Air Campaign (TAC)-SAGE model which determines the theaterwide allocation of "red" and "blue" aircraft under combat conditions. The decision process occurs over a specified number of days of conflict using ground force movement as a measure of merit. Many projects have employed this model, and with each, enhancements were made to increase its scope and generality. In particular, the embedded ground war simulation has been redone a number of times to increase fidelity. There are at

[1]Although neither the SAGE algorithm nor the TAC-SAGE or T-SAGE models have been documented in releasable publications, a variety of studies have and are using the model. The primary model designers and developers are Richard Hillestad and Louis Moore at RAND.

least four versions of the original TAC-SAGE model and two major recent versions which include more detailed ground simulations and explicit air allocation algorithms. An attempt was made to incorporate the SAGE methodology into a combined air and ground allocation model called Theater (T)-SAGE. The current, ongoing effort related to SAGE is a refinement of the methodology and implementation strategy into a new model called Theater Level Combat (TLC).

The SAGE technique is, for many RAND projects, the method of choice for performing quantitative theaterwide combat analysis. However, the operational complexity associated with applying the SAGE family of models prohibits their overall usability. The major problems center around database management (i.e., being able to create and manipulate the large amount of requisite data that is representative of the problem under examination and consistent with the model structure). Interpretation and portrayal of the results is also a very resource-intensive task because of the vast quantity and the complexity of data involved. Input data are currently gathered mostly by hand from a variety of source databases and the results of other models. This process can take upward of a man-month to consolidate the information and insure its structural validity. There are far too few members of the research staff who can successfully derive these data files, and their availability is severely limited. Furthermore, it is difficult to find others who will take on these tedious and pressured tasks.

Performing sensitivity or excursion analysis appears to be a fairly simple task, but because of the data structure and detail, the analyst must rely on these same model experts. The overall manpower requirement for a particular study can be between one to three man-months just to organize the information and produce the appropriate displays. These costs do not even begin to address (1) the opportunity costs associated with delays to the research agenda, (2) more creative analysis the model experts could be pursuing, (3) errors generated in the modeling process which affect the corresponding analysis, and (4) the severe underuse of computer power. Because of the virtually independent nature of most studies, these costs recur with virtually no sharing of improvement/enhancement costs or reusability of the "lessons learned." Although most project leaders agree to funding the marginal cost of specific model improvements, few wish to pay for the integration, upgrading, maintenance, and general enhancement costs. Without question, information processing and the data manipulation tools that provide direct model access to the analyst, without requiring the interactions of model experts, are vital.

## NOT JUST SAGE

The problems noted above represent only a subset of the frustrations analysts must face when undertaking research that incorporates models and simulations. These difficulties are not peculiar to the SAGE models but are descriptive of virtually all major internally created or externally obtained models at RAND. After many lengthy discussions with modeling experts outside of RAND, it is clear that these types of problems plague the modeling community as a whole. A recent article on scientific programming points to three major deficiencies in developing "home grown" models — limited documentation, lack of flexibility/modularity, and poor overall design (Dazzo, 1988). These are common problems wherever modeling is performed. Whether the term "model" refers to a collection of algorithms, a closed analytical solution methodology, or an extensive simulation, these problem characteristics appear to be universal.

Some might wonder why these difficulties are at once easily recognized and yet unresolved. Part of the reason is because the model, particularly in the RAND context, in and of itself is of little value. Even where the model is a marketable product, its existence is justified by the quality of its operation, usability in the decisionmaking process, and ultimately the "real" world impact of the results it produces. Thus, just enough resources are typically allocated to the modeling process to achieve results and maintain the model as a "tool." To make significant improvements in the overall modeling environment, it is necessary to develop support systems for the model — that is, a set of tools for *the tool.* Since resources allocated to a particular model are rarely abundant, and in reality often inadequate, it is easy to see why there is a deficiency in the resources, as well as attention, given to the development of a more general modeling environment.

The lack of modeling support systems is aggravated by the fact that building a robust environment is a difficult task, requiring the dedication of a large number of assets (specifically, people, money, and time) and, particularly, a talented, visionary staff. Developments of such tools in the past (e.g., data management, graphics, spread sheets, and algorithmic libraries) have often not lived up to the expectations of the user community. Not only are there frequently missing functional elements in the software, but the systems are often error prone. Although many pieces of the modeling puzzle exist in the community, developing and gaining acceptance of standards also precludes the establishment of a more flexible modeling environment.

Individual model restrictions are only accentuated in a multiple model environment, such as that at RAND, because no two models possess the same support or communication mechanisms. This means that two or three experts dedicated to *each* model are required across the board. Some models include various support systems that reduce manpower needs, but because of the specific nature of their implementation, these support systems are nonstandard, nontransferable, and, yet, functionally repetitive. For example, within one model, study-critical support features may be implemented that are simply unavailable in other models required for the same study. Furthermore, if a feature is available in multiple models, it is undoubtedly implemented inconsistently.

## THE MODEL INTEGRATION AND MANAGEMENT SYSTEM

The purpose of this research is to expand the state of the art in model support systems by defining a conceptual master environment for implementing, utilizing, and maintaining existing, imported models as well as providing many of the common building blocks for developing new models. The motivation for this effort has primarily evolved from attempts to provide analysts with a simple, consistent environment for performing quantitative analysis. This Note describes the conceptual design of an automated, flexible, and general purpose Model Integration and Management System (MIMS) intended to relieve the analyst and model experts of many tedious modeling-oriented tasks and to incorporate a robust suite of analysis tools.

The MIMS project does not involve the creation of any new models but rather is the development of a state of the art modeling environment by which analysts may prototype, implement, and maintain models and their associated databases for policy-oriented research in a rapid, efficient, and consistent manner. The key methodology is to provide the analyst with high-level decision support modeling tools and to build an "expert system"-like environment using intelligent databases to replace the technical analysts and programmers who traditionally provide the interface between analyst and model.

The MIMS is a total-system-solution methodology employing a number of advanced software tools, purposely designed to be responsive to the variety of modeling deficiencies noted above. The underlying knowledge-based system requires explicit descriptions of source and model databases. Once this is accomplished, the system can provide the user with a comprehensive set of data-oriented documentation. Modeling tools and subsystems can be sequentially developed and generated for individual functions across models, instead of for each model. The framework is inherently modular and will allow for fundamental

development activities and the creation of rudimentary tools to occur over time. The functionality can be developed early without sacrificing eventual system robustness. The prototype application for the MIMS will be a stable version of the SAGE model. The particular version of SAGE chosen is not significant. The whole purpose of the MIMS is to provide a rapid implementation, use, and analysis environment. Once the MIMS is in place, any SAGE version (or multiple versions) will be easy to implement, as will any other model required at that time. Lessons learned from modeling and simulation activities within RAND's Military Operations Simulation Facility (MOSF) will be incorporated into the conceptual and actual development of the MIMS.

The next section provides more detailed background on the current state of modeling and reviews the effort required for applying models. Section III summarizes the analyst's perspective in modeling and the set of tools required to more easily perform quantitative analysis. Section IV appeals to the constraints of the model and describes the wide communication gap between analyst and model. Section V specifically focuses on the MIMS. The methodology and system structure are described and related to specific developmental tasks. A phased, coordinated implementation plan is also discussed. The final section summarizes the expected utility of the MIMS and suggests potential extensions.

## II. WHY IS APPLYING MODELS SO HARD?

The use of computerized models and simulations to examine complex decision and policy issues has been well established and dates back to the advent of the computer age. Particularly in military-oriented analysis, modeling has been used to gain insights into large-scope, extensively detailed issues. Solving relatively simplistic numerical problems has evolved through the years to incorporate sophisticated computer systems that explore intricate design and operational capabilities of individual entities as well as the complex interactions of numerous tactical and strategic systems. Typically, the analyst wishes to use a model or simulation to quantify the most significant benefits or costs of a new system, such as a weapon or sensor, or to view the effects of a change in policy or doctrine. The model is implemented as a decision aid to extrapolate from observed data or to provide insights into the potential of intangible alternatives.

In this section, the two modeling environments commonly used today are described with respect to the three principal tasks embedded in the modeling process. The operational "architecture" or methodology is presented to provide a framework for understanding how the process is accomplished. The first of these is the "brute force," unsupported environment. Some improvements are gained with the second or semi-automated environment, but it is shown that neither of these architectures adequately resolves the analyst modeling needs raised in Sec. I.

### THE UNSUPPORTED ENVIRONMENT

Direct analyst interactions with models have traditionally been difficult. The advances in computer software and hardware engineering have provided the analyst with enormous capabilities for performing policy analysis but have created special challenges by requiring expertise in database management, computer graphics, mathematical algorithms, interprocess communications and protocols, and computer languages. To perform these tasks, the analyst insulates himself with a computer-oriented technical staff who work with the model components.

Figure 1 displays the architecture of this very typical operating environment for most models involved in performing quantitative analysis. In this and other diagrams, the boxes represent information sources and the ellipses are computer processes. The analyst, as well as the database and modeling experts, are shown in italics and dotted boxes to indicate their

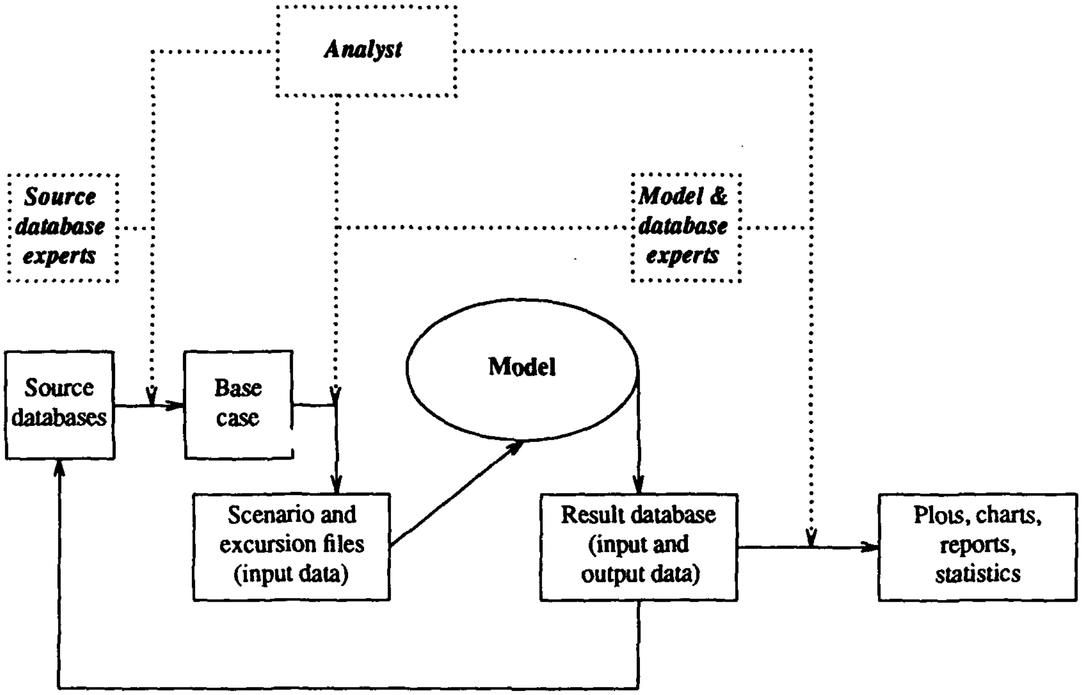


Fig. 1—Typical unsupported model environment

interactivity with the computerized modeling environment. The lines and arrows indicate the flow of information. Solid lines depict the actual flow of data in the modeling process, and dotted lines represent "manual" or nonautomatic data processing in which a person intervenes or interacts with the data flow.

In organizing information required for the model, the analyst must extract relevant data from appropriate sources, apply the model to examine alteratives and to test assumptions, and create descriptive and summary materials to present to a specific audience. The analyst must work closely with the model and database experts to insure that compiled information properly reflects the desired assumptions and preferences to be incorporated into the modeling effort. The general tasks involved in this process (source data preparation, model database creation, and model results analysis) are depicted in Fig. 1 and are described in detail below.

**Preparing Source Data**

The first step (and challenge) for the analyst is to derive the best estimated data representation of the systems to be studied. This is shown in Fig. 1 as the arrow from the Source Database box to the Base Case box, and includes the manual interactions by the analysis and database experts. The goal is to extract and to manipulate a variety of source databases into the analyst's perspective of the "world," based on the study requirements. This might include descriptions of objects (e.g., headquarters, units, and targets), environmental features (e.g., terrain and weather), or performance factors (e.g., movement rates, attrition values, and probabilities of kill). The Base Case should be the smallest information set that encompasses a comprehensive description of all elements at the appropriate level of resolution needed for the particular study. Additionally, our definition of Source Databases includes the results of other models or simulations (shown in Fig. 1 as a "feedback" data loop from the Result Databases of one or more models to the Source Databases of another).

Many steps are required to adequately synthesize source data. Information on potential suppliers or a plan for data generation must be determined. These Source Databases can include orders of battle, targeting databases, equipment lists, environment databases, and tables of system characteristics. Once obtained, the source data must be cross-checked for consistency, subset to eliminate unnecessary data, aggregated to reduce detail, and generalized to achieve the proper scope for both the study objectives and the models involved. The analyst must supply the appropriate assumptions for dealing with missing or incomplete data and information that is inconsistent either within a database or between the variety of source databases under examination.

Most of the data preparation tasks mentioned so far deal with only a single database or model. Integrating several databases or preparing data for multi-model analysis is many times more difficult. Rarely are two databases or models constructed with even similar assumptions. Conflicts in overlapping data sources or requirements must be resolved. For example, slightly different assumptions might lead one source to indicate that the primary role of a certain airfield is to support ground attack aircraft, whereas another source will assume that the primary role is to support air defense aircraft.

In performing the data preparation step, the analyst must interact frequently with database experts not only to properly interpret the documentation, organization, structure, format, assumptions, and values in the source databases, but also to assess the reliability or appropriateness of data items. Control of database versions and updates must be readily

handled to avoid use of outdated information and to enhance the accuracy of the data to be maintained. Only by a team effort between the database experts elucidating database assumptions and the analyst expressing study requirements can this step be successfully accomplished.

The current generation of tools that enable an analyst and source database experts to perform these tasks is severely limited. Recent advances in multipurpose database management systems have eased the organization and standardization issues associated with database maintenance, but in general, these systems are not robust enough to handle the great variety of forms, structures, and hierarchies found in large military databases. Analysts and database experts must rely on rudimentary text editors or they must write specific data manipulation routines to perform data management tasks. In addition, tools for incorporating descriptive and semantic information including interactive documentation, consistency rules, and assumptions are not yet available in commercial data management packages. These enhanced data management considerations are a fundamental part of the MOSF research agenda and are also being explored by the Intelligent Database Project (Cammarata, 1988).

Before leaving the data preparation step, a more detailed explanation of the Base Case is in order. To some, the creation of the Base Case may seem an unnecessary part of the modeling process. Rarely is this step explicitly defined, but frequently it is performed, perhaps unintentionally. Source Databases are frequently too awkward (that is, individually too large or jointly too complex) to manipulate directly into a model. The Base Case is explicitly defined here because it not only represents a crucial step in the modeling process, but will also be shown to be a significant element in the automated modeling environment discussed in Sec. IV.

The data collected into the Base Case may be prepared for a variety of models. Portions may have little relevance to a particular model, but are desirable for direct data analysis or to provide other insights for the study. As study objectives broaden, the Base Case can be increased to include new or refined information. Indeed, the Base Case is a collection of all the data needed for a specific study that may use multiple models, each requiring a specific subset of the Base Case. Data representing the appropriate scope and aggregation for each model to be used must be included in the Base Case. However, the Base Case is not just a model database. Each model requires only those parameters necessary for its particular operation (i.e., print modes, model control, initial states, and algorithm parameters). These parameters are selected for particular model scenarios or excursions and should not be included in the Base Case.

As previously mentioned, one purpose of the Base Case is to avoid cumbersome, unwanted elements in source databases while retaining the original source databases intact. A fundamental misconception often held by analysts manipulating source data for a particular project is that they may freely manipulate, replace, and delete source data items because their perspective or assumptions apply universally. Too often, another project wishes to examine precisely those data items that have been replaced or deleted by a prior project. The Base Case provides a physically separate area for the project to accomplish the data processing and integration tasks required without endangering source data items.

Finally, the Base Case is a useful intermediary form, between the source data and the model input, that can be structured in an appropriate format, specifically for the analyst, and thereby can eliminate the peculiarities of both the source and model database formats. All too often, the Base Case is stored in a form that either emulates the original source data or is similar to the required model format. Unfortunately, neither construction is convenient for the analyst. With the appropriate support tools, the Base Case may be defined in such a way that the analyst can more easily extract Source Data into it and create model databases from it. This concept is further explored in Sec. IV.

### Creating Model Databases

The second step for the analyst is to prepare the actual data files, called "Scenarios," which will be read by the model. This step is represented in Fig. 1 by the arrow from the Base Case box to the Scenario and Excursion Files (Input Data) box with manual processing being performed by the analyst and by model and database experts. Model operation parameters are combined with the Base Case data and transformed to the specific format required for the model. These parameters can include display or printing options, functional modes, configuration values, and, for sophisticated numerical models, such data as convergence parameters or initial algorithm conditions. The analyst must work closely with the model expert to derive these data items to insure proper model operation and results.

Analysts may also alter modeling or data assumptions inherent in the Base Case for at least two reasons. First, possible alternative policies may be examined directly by creating excursions on the Base Case. This approach is often used to explore "what if" possibilities. An example is the selection of an alternative system, like a communications network or sensor, or the implementation of a different strategy, such as attacking in a different location. Second, source data and model content alone are never sufficient to predict the behavior of the actual "world," and hence, sensitivity and parametric analysis is performed to test how

critical certain assumptions or estimations are to the stability of the model results. An analyst might want to examine the results of altering a kill probability uniformly around its mean, say from 0.6 to 0.8 in steps of 0.05.

The lack of model support tools has prompted many model designers to include a variety of input data management options bundled together with the actual unique computational or algorithmic processes performed by the model. These model-specific operational modes are usually difficult to manipulate and typically apply only to specific applications of the model. For example, the TAC-SAGE model contains parameters to augment sets of attrition rates or probabilities of kill. These were established to perform certain types of sensitivity analysis, but are not universally available and are inconsistent in their positions within the data stream and their interpreted functional effects. Furthermore, helpful functions embedded in one model will not be implemented in others, or will be implemented using different syntax, protocols, devices, or interfaces.

Perhaps the greatest difficulty involved with an extensive model database is to maintain the proper consistency throughout the data sets. For example, if an analyst adds an aircraft type to TAC-SAGE, there is no automated mechanism to provide the full set of other data items that also must be added or modified. Many times, even a small, seemingly insignificant change to the database may require additional elements to be modified in value or in format. An inconsistent or incomplete data set will presumably cause the model to abnormally abort. However, some modifications may be accepted by the model only to produce erroneous results. Without automated tools to warn of these conditions, the analyst may lose valuable time, or, worse, be led to accept false results.

### Analyzing Model Results

After the model processes the scenario and excursion files, the final task is to interpret and analyze the results. Typically, this information must be condensed, usually into a graphical or tabular form for presentation to a larger audience. In Fig. 1, this step is shown by an arrow from the Result Database (incorporating both input and output data) to a box annotated by Plots, Charts, Reports, and Statistics. As before, a dotted arrow from the analyst and model and database experts indicates that manual intervention is required to perform this step.

Result files created by the model are often formatted for computational convenience and not for display or presentation. Here again, the analyst must work with a model expert to properly interpret the processed information. Data must be reconfigured, and specialists in data management, statistics, and graphics must be involved to prepare data in a finalized

form. Typically, special software routines are written to interface with printing and graphics devices. For example, TAC-SAGE has been extended to access special types of graphics devices for display of particular information. This process is tailored for subsets of result data and lacks generality. In a multi-model study, the analyst is frequently frustrated with the rigid mix of required hardware, each model being accessible through one set of devices. Either the data must be "carried" from machine to machine, or the models must be ported to a common set of devices. Neither of these solutions is satisfactory. The first creates serious delays and the potential for error. The second is an enormous undertaking particularly if specialized software has been used.

A primary purpose of the modeling process is to provide the ability to assess alternative assumptions, tactics, systems, configurations, or capabilities as well as to derive the sensitivity of various elements in the analysis. Without an automated environment to manage cases or excursions, piles of model output or megabytes of disk space are left for an individual to tediously examine and synthesize. This manual approach requires a well trained person, otherwise, crucial and often subtle differences or trends will be overlooked. Moreover, it can be easily recognized that not only is this a poor use of resources, but who would ever want to volunteer to perform such arduous tasks? Hence, the very important matter of comparing cases is often left undone or is not sufficiently done.

**PRE- AND POST-PROCESSORS, A SEMI-AUTOMATED ENVIRONMENT**

In recent years, emphasis has been placed on automated tools for easing the analyst and model database expert's manipulation of source and scenario files, result databases, and display devices. Figure 2 portrays the data flow in these more automated systems. Pre- and post-processing routines are added as tools to supply at least some standardization in the data manipulation *for the particular model*. In many cases, the pre- and post-processors dramatically increase the usability of the model. The analyst does not need to rely as heavily on model and computer specialists, and these specialists are free to work on other, less mundane tasks.

However, these methods for model support are not panaceas. The processing of Source Databases remains as tedious as before, only the model databases are easier to manipulate. The embedded routines provide data management and graphics support, but in a nonstandard form. Thus, if an analyst wishes to perform a certain operation, such as draw a plot or modify a parameter, the syntax, as well as the way to request and execute this function, will undoubtedly be different from model to model, if indeed it can be done at all. Furthermore, the pre- and post-processors are usually machine-dependent requiring

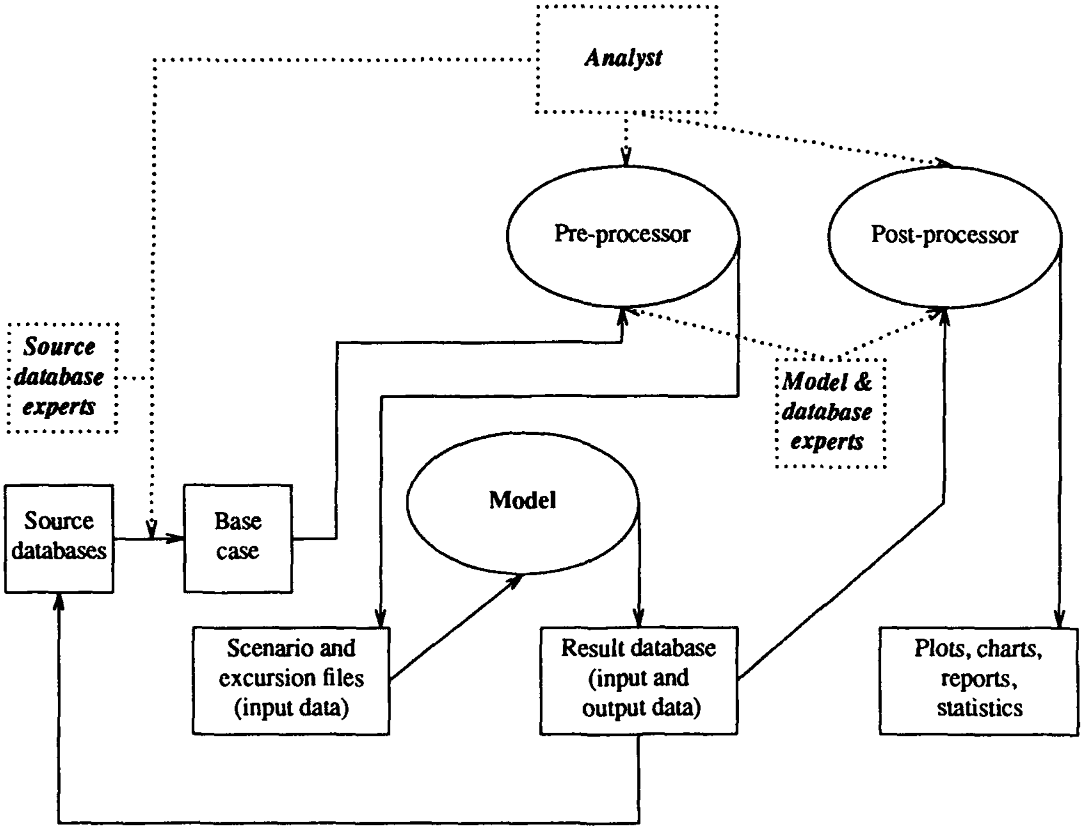


Fig. 2—Current semi-automated model support environment

specialized display manipulation or other system-specific functions. If a model is to be expanded to a new environment, tedious code enhancements are required. An important consideration is the overall opportunity costs associated with adding functionality model by model rather than to the system as a whole. The analyst may have had some of the preparation burden relieved within the context of a single model, only to be bogged down all the more within a multiple model environment.

## III. THE ANALYST'S PERSPECTIVE

To understand the high-level requirements for the MIMS, it is necessary to gain an appreciation of the analyst's frustrations in applying models. A vast range of individuals can be considered as analysts. We place no qualifications on the type of research activities or applications, administrative responsibilities, or technical skills making up the analyst's workload. Only a desire or need to use a model is significant.

The analyst's need for modeling stems from an underlying research objective. The model provides a quantitative structure and a standard means for evaluating various alternatives. Although the model is an idealistic representation of the world, it is designed to consider those elements most critical to the research activity. Additional "expertise" can be built into the model as time or testing demonstrates deficiencies in the scope or breath of elements within the model. Sharing models that have obtained credibility in a particular application area is also important to the analyst. Studies employing the same model can more easily be compared. Moreover, the extensive use of a model is one of the surest ways to eliminate errors.

In practice, tasks associated with model application require extraordinary amounts of time and effort. The intricacy of the operation promotes delays and allows errors to be introduced. The analyst is faced with having to pioneer the use of the model, databases, and the supporting software (for which he often has little expertise, time, or patience) or the analyst must engage a group of experts to perform these tasks. Misunderstandings between the analyst, model and database experts, and other technical support people are accentuated with the delays associated with preparing the model for operation and the model operations that produce results. The analyst must either assume the role of an administrator (which reduces the time available for performing the analysis) or allow modeling activities to proceed without sufficient direction. Each model requires at least two or three dedicated people to initialize, operate, and maintain the databases and the model. Many frustrated analysts will attest to the fact that a model cannot be left dormant for any significant length of time and then be expected to function properly. In short, although the *ability* to efficiently perform high-quality quantitative analysis exists, the lack of a structured, model-independent support environment severely limits the *actual* analysis that is performed.

The solution for these problems is straightforward: the analyst must be given a single set of automated tools that assist in performing data and model manipulation tasks rapidly and interactively. Functional operations should be activated naturally, as extensions of the analyst's instinct. Functions with source database integration include comparing data items, aggregating values, joining data items from multiple databases, and simplifying "hierarchical" and "ownership" relationships. To prepare data for a particular model, the analyst must create specific instances or scenarios, modify data for excursions or parametric analysis, perform experimental design to test alternatives, assess data consistency, and determine completeness for a particular model. Model results analysis requires controlling cases and excursions, supplying reports and charts, creating plots and graphics, and performing a variety of statistical tasks. The man-machine interface should be structured so that both novice and expert users are satisfied. The "user-friendly" interface should be informative and attractive, inviting use. Beyond user-friendliness, one consistent set of protocols should be developed on the basis of functional operations, fully independent of the model. Although it is unrealistic to assume that a modeling environment can be predictively designed and built to perform every task that every analyst desires, it is possible to define a robust, high-level environment where a large portion (say, 75 percent or more) of the analyst's functional requirements are included within a modular, expandable support system of uniform tools.

The type of capabilities described here for the analyst largely resembles the role of a Decision Support System (DSS) (Sprague and Carlson, 1982). A DSS is an automated system that enhances a decisionmaker's judgment by rapidly providing information from models or databases. The purpose is not to replace the decisionmaking process, but to provide sufficient information for the decisionmaker to arrive at a more intelligent, deliberate decision. The DSS provides direct user controls, includes a "tool box" of analysis and evaluation resources, and is adaptive to extensions. Typically, these types of systems have been implemented for aiding financial decision processes where the problem structure is well defined and the information flow very broad. The DSS concept resembles the type of high-level analyst interface required in the modeling process, in that it is designed to provide the user exactly what is wanted out of the system. However, the modeling/analyst process is significantly more complicated and broad, since general modeling tasks are much more unstructured and the related analysis much more ill-defined.

In the best of all worlds, the analyst would prefer to perform much of the quantitative analysis directly, if a set of tools were available for efficiently performing fundamental operations. The interfaces to these tools must be simple enough for the novice to learn and ultimately extendable to the full set of available functions. Figure 3 depicts the three general steps of quantitative analysis that the analyst would like to perform more efficiently. It is the analyst's responsibility to determine how source data should be integrated to form a Base Case representation for the study. This does not mean that the analyst should worry about data organizational details (e.g., formats, interrelationships, and structure). These details should be automated through "data integration" tools so that the analyst's time can truly be spent synthesizing the data. Similarly, the analyst provides the decision framework from which alternatives, scenarios, and parametric analyses are carried out in the modeling effort, but scenario generation tools should be available to eliminate the need for a detailed knowledge of the model's databases. Finally, the tools to examine the results and prepare suitable presentation media should be available to the analyst so that the bulk of the analyst's time can be spent examining the critical issues supported by the modeling effort, rather than porting data and models from one system to another in search of the desired software tools or output devices. This approach provides the analyst with greater opportunities to perform the actual analysis thoroughly and efficiently.

Clearly, the set of support tools for the analyst must be contained in a system that is rapidly expandable. A common complaint from project administrators and analysts is that the investments made in developing software modeling aids can never be fully taken advantage of. Too many critical resources are spent on software from which little is ever received. Historically, this has been the case because functional capabilities have always

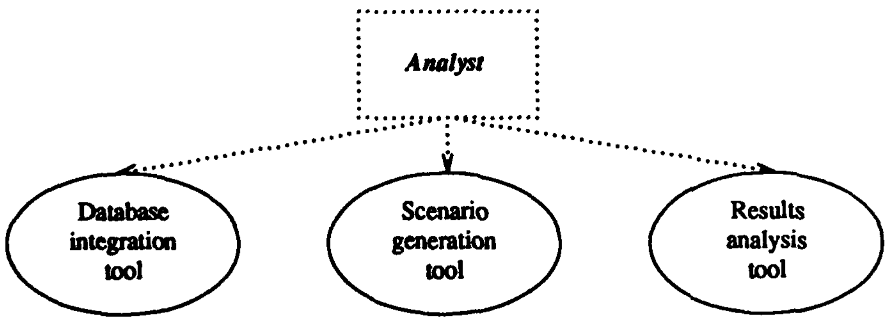


Fig. 3—The analyst's perspective

been embedded into the specific model, and hence are nontransferable to other models or applications. The analyst's perspective includes the desire to achieve operational capabilities (such as access to a specific graphics device or database manipulation technique) once for all models within a standard, uniform set of interfaces and protocols. With this as a fundamental requirement for a modeling support system, the additional advantage of being able to include functional capabilities piece by piece is also realized.

It is particularly appropriate to point out here that the analyst's perception of how to use the model need not (and should not) be encumbered with the physical details of how the model actually is executed or how the modeling support system is linked to the model. Analysts have little desire to understand the intricacies as long as they are confident that the system is operating properly. However, these details are discussed in this Note to provide a comprehensive view of the MIMS. The next section describes the operational characteristics of data preparation, model operation, and results analysis. Section V provides the conceptual design of the MIMS which intervenes between analyst and model so that both may operate in their own preferred environment.

## IV. MODELS: RIGID SOFTWARE

Although the analyst's needs are satisfied by providing the tools and interfaces required to enhance model operating efficiency, the underlying data processing tasks described above must still be performed. Figure 4 reminds us that source data must be synthesized into the Base Case. The Base Case must be formatted for model processing. Model results must be compiled into useful presentation forms. The only help available to perform these tasks includes database and model experts, as well as various handbooks and documents that describe the data. The obvious question is: How do we provide a connection between analyst and model that takes full advantage of the expertise available?

The natural first approach is a *brute force* solution in which tools are created and integrated within each model in an "as needed" or "ad hoc" manner. The advantage to this method is that requisite resources for making these improvements are allocated in response to specific needs and can be related (and charged against) specific studies. There is no need to justify the funding for a long-term development activity whose benefits may not be realized until after current projects have ended. However, this approach has no long-term focus or structured application base and is potentially as resource intensive as the environment that currently exists. Changes can be made for large, long-term projects, but "smaller" projects can rarely afford the outlay of already constrained resources. Furthermore, very little synergistic effects can be realized from these kinds of enhancements, since a sufficient level of generality is rarely attained.

Another common approach at the other extreme is to develop a richly supported environment that requires models to be recoded for the system. The benefit of this method is that all the models share the same standards and are internally compatible. However, this approach is also unsatisfactory because a great deal of capabilities already exist in "noncompatible" languages, analyst tools, and hardware. It is desirable to exploit this software "capital" rather than insisting that it all must be translated or rebuilt into the new environment. Rewriting software (models or other analyst tools) is a high-risk operation often requiring talented technical people whose time is already in short supply. Experienced modelers know that even seemingly independent or unrelated alterations to source code, such as adding a "print" statement, can produce catastrophic results. It is not hard to imagine the potentially disastrous effects of embedding major data management or graphics routines within the model or requiring a full rewrite into a new environment.

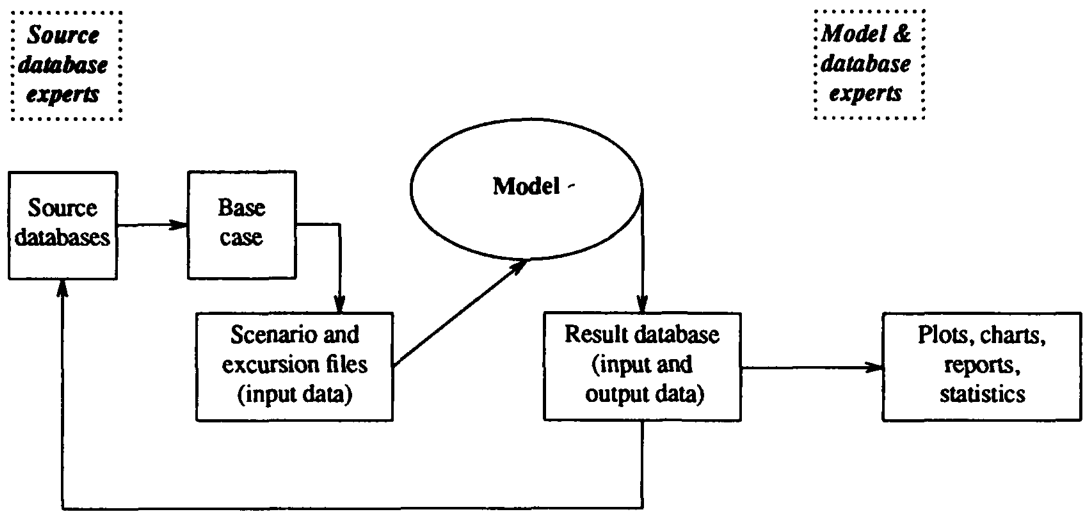


Fig. 4—The model's needs

Herein lies a subtle paradox. Models are coded in software, but are far from "soft" or pliable. These codes are soft only in the sense that modifications can be easily performed. However, there are never any guarantees that the code will be amenable to these changes. Models are ultimately rigid, since it is nearly impossible to determine if code alterations are appropriate or correct, particularly in the context of the model's purpose and the model's other functions.

All of these issues argue for a modeling environment where information and data processing, model use, and analysis functions can be added to the model software externally without requiring the model itself to be altered. Thus, the model is comprised of the unique computational aspects rather than being bogged down with information processing tools. The model developer can be a modeler and leave data management to information scientists and graphics to graphical engineers. Furthermore, the analyst can focus on determining the appropriateness and effect of the model rather than being encumbered by computer-oriented details. Being able to rely on the automated system also means that the analyst may more confidently determine the extent and depth of the modeling analysis that can be performed.

Exploiting the implicit features of the Base Case helps to facilitate this modular modeling environment. Let us again examine the Base Case by noting the rigid structure of the other four information sources (the solid boxes in Fig. 4). Source Databases are determined exclusively by external data sources. Typically, these databases are constructed

archaically (using "card" formats), more appropriately for data collectors than for data users. Model databases (both input and output) are equally anchored and are often awkward to manipulate, either containing binary data or organized in a puzzling manner that can only be explained by computational convenience or by the historical evolution of the model. Finally, device-oriented data are structured for device efficiency and are similarly rigid and equally impenetrable.

Unlike the other information sources in the data flow, the Base Case format, structure, and organization can be constructed in any appropriate manner. It can be as poorly defined as the other data forms, or it can be exploited to provide maximum accessibility by an analyst. This observation has significant implications for an improved modeling framework. The Base Case should incorporate two important characteristics. First, it should be in a physical format that can be readily accessible to a data management system. Second, in addition to the actual data, provisions for descriptive information should be included to enhance the interpretation, understanding, and methods for presenting the data. This information includes descriptions of data items and relationships, explanations of codes, and specifications of hierarchies or ownerships. The Base Case format can be standardized to increase the use of manipulation tools generally available.

# V. THE MODEL INTEGRATION AND MANAGEMENT SYSTEM: BRIDGING THE VAST GAP

The MIMS provides an automated environment that includes decision support elements for the analyst, significantly reduces the extensive requirement for model and database experts, and maintains without model alterations the flow of data from source databases through the model to output devices. As noted in the last two sections, a large discontinuity exists between the desired operating environment for the analyst and the rigid implementation requirements for the model and associated databases. To "bridge" this analyst-model gap, while reducing resources needed currently to perform the interface, the MIMS must be innovative and flexible.

Figure 5 shows the current gap between the analyst and the model and data with the process experts positioned in the middle. The key realization is that these data, model, and software engineers are indeed the experts, and that the function they provide in the modeling effort is often systemic, based on an intense set of rules and human knowledge. Indeed, a natural application for advanced software engineering is to replace the model and database experts with an *expert or knowledge-based system* (Hayes-Roth, et al., 1983) where the data rule systems are implemented and maintained in *intelligent databases* (Cammarata, 1988). The following brief description of knowledge-based systems is provided to acquaint the reader with some of the general concepts that apply to the MIMS.

## KNOWLEDGE-BASED SYSTEM AND INTELLIGENT DATABASES

The term "knowledge-based system" refers to an automated environment where systematic elements of human expertise are replaced by a corresponding software system that solves some of the same real-world problems. In the modeling process context, the human expertise is embodied in a number of model and database experts, as well as a variety of handbooks and manuals. The knowledge-based system is used as a structure for manipulating this information and bridging the analyst-model gap. Special emphasis is placed on "expert knowledge," which includes a full syntactic and at least a partial semantic description of the source and model databases. Because of the high degree of flexibility stipulated in our desire to integrate *any model*, a robust data-structuring mechanism employing intelligent database methodologies is required to support the knowledge-based system.

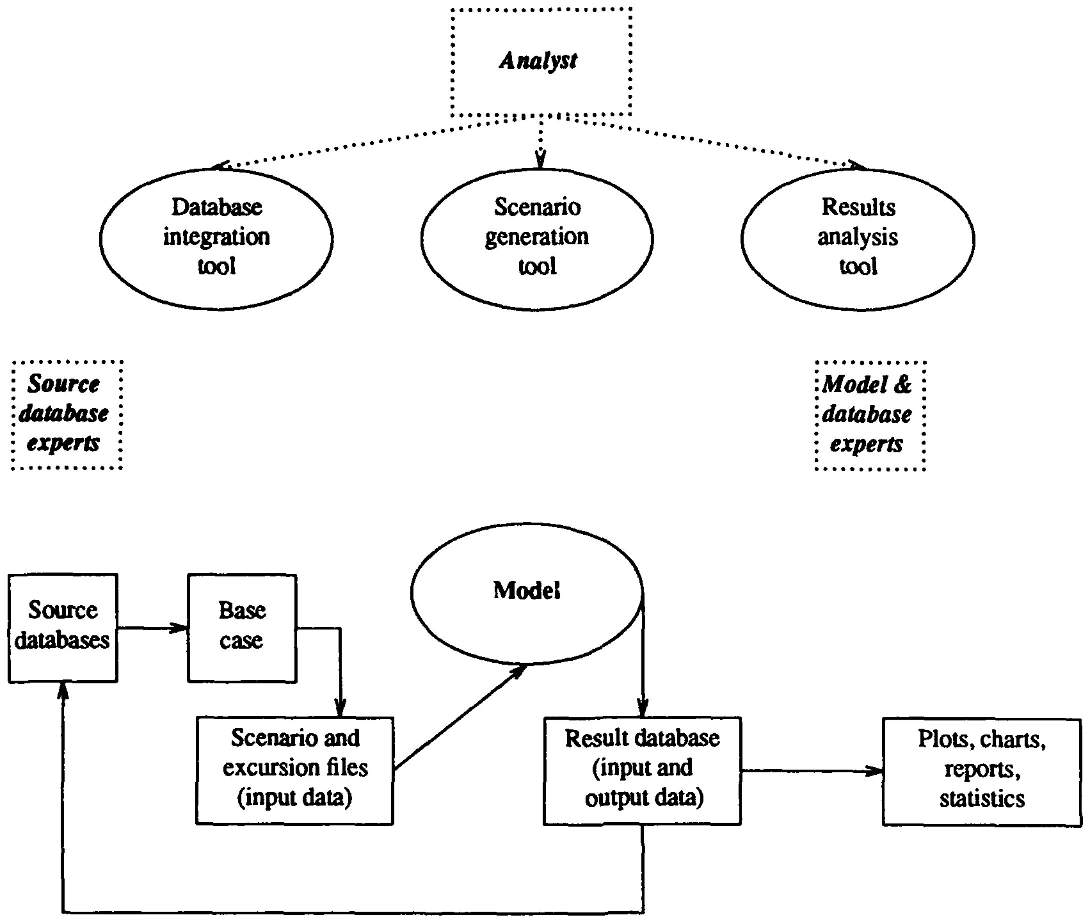


Fig. 5—The analyst/model gap

Figure 6 depicts our analyst-oriented architecture for the MIMS. Although more complicated than previous diagrams, boxes still represent information sources and ellipses are computer processes. Data flow is shown with solid arrows and manual intervention by dotted arrows. Additionally, dashed arrows are used to represent "meta" data; that is, data that describe and interpret actual data. Letters annotate command or instruction paths, and numbers indicate the data flow paths.

The MIMS diagram is explained below from three different perspectives. First, the data flow from source databases through the model to output devices is described. Second, examination of the operational levels of the MIMS highlights received benefits. Finally, looking at the analyst functions provides the structure for how the MIMS should be developed.

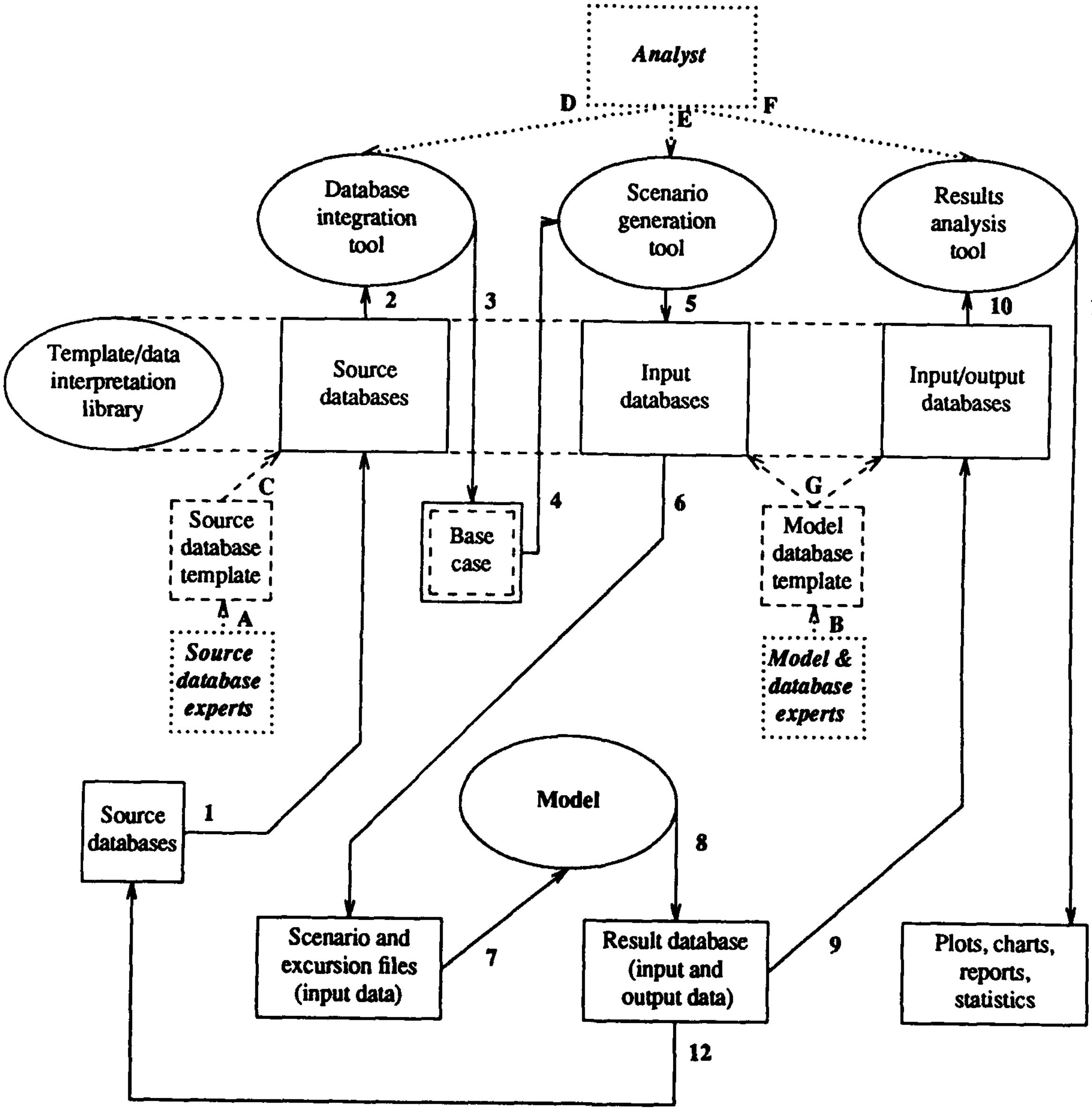


Fig. 6—The MIMS

**DATA FLOW — HOW THE MIMS WORKS**

Beginning with a new source database or model, the first step is the installation process. This means that the syntax, format, relationships, references, and descriptions of both the source and model databases must be extracted from experts and placed into a robust database template (shown as A and B, respectively, in Fig. 6). This template, one for each database or model, is a schema for reading and interpreting the actual data. For models, the template contains a description for both input and output data. The database template incorporates intelligent database methodologies to capture a comprehensive, detailed schema of the information now only in the minds of database and model experts or in the volumes of associated user and analyst manuals. Although the creation of the template is an arduous task, once done, it can be widely used and easily updated. Furthermore, the template is the ultimate documentation for the actual database. With the requisite source and model data templates defined, the flow from source data to model results analysis can be accomplished. In presenting these steps, the alphabetic or numeric path label from Fig. 6 is given in parentheses.

Data integration is initiated as an analyst selects a database for manipulation. The associated source database template provides format transformation and conceptual structuring information to the template/data interpretation library (C). This library makes up the heart of the knowledge-based system and performs two functions. First, it reads and interprets the source data (1) changing the physical structure to a MIMS database management system (DBMS) format.[2] Second, the library transforms the source database into a preferred logical structure (2) that is standardized for the variety of data integration tools.[3] This latter format, which can be used with all of the analyst tools, is referred to as the model support system (MSS) format. The analyst can then aggregate, generalize, subset, and integrate these data (D) with a consistent set of tools. Note that these data integration

---

[2]With the inclusion of a modest amount of additional information, the template can also be used to restore the source data from the MIMS DBMS format. In this way, the original data can be directly compared with the transformed data to insure that nothing has been modified or omitted. Furthermore, other systems may require the data in the original format, which the MIMS can then readily provide.

[3]In this Note, the formal structure and design of the template and the preferred physical and logical forms are purposely left unspecified. Indeed, these are part of the research and development tasks of the MIMS. A possible example of the physical or MIMS DBMS format is one that is amenable to an object-management system. The template would then include rules or methods for transforming the actual data into the object-management system. The logical form would then be object-oriented, incorporating information from the template as to the relationships between objects. The template should also include the preferred means of presenting the data organization and structure.

tools include browsing, statistical processing, plotting, and geographical or geometric portrayal which help the analyst perform this task more accurately and efficiently. These tools will give the analyst the ability to resolve missing data, inconsistencies, or any other peculiarities. Multiple database integration and cross-comparison tools are supplied. The analyst can also incorporate study and data assumptions or insights gained about the appropriateness of certain databases and data items. When the analyst is satisfied with the content of the data and the related semantic and descriptive information, this entire information set is stored into a Base Case (3).

Because the Base Case is an intermediate or neutral information set, it can be readily structured in the MIMS DBMS format. In Fig. 6, the Base Case is raised from the model data flow level, depicted on previous diagrams, and shown on the same level as the database and model templates. The primary reason is that it combines both data and meta-data, containing its own template. Therefore, the Base Case is represented by both a solid and a dotted box.

Scenario generation is initiated by the analyst's selection of a Base Case (D) and the automatic flow of this information to the support system tools (4). Model specifications are added, cases defined, and parametric analysis of excursions determined by direct analyst interaction. To transform this data into the model format, the model input data template is processed by the template/data interpretation library (G). Again, a two-phased transformation occurs to produce the model input database. First, the logical structure is manipulated with particular attention to the correctness and completeness of the data necessary for the model (5). If inconsistencies or missing elements are found, the analyst is notified and required to make the appropriate adjustments before the modeling process continues. Second, the model physical input format is created (6). With the input files available, the model automatically manipulates the input data (7) and produces the associated output files (8).

The final step in the modeling process is to generate suitable output products. These include plots, charts, reports, and statistics related to the cases run through the model. The model database template supplies the template/data interpretation library (G) with the information needed to read the model database (9). These data are transformed to the MIMS physical and logical formats (10). The analyst uses a uniform set of results analysis tools (F) to perform case control, manipulate the data, and process this information into the appropriate display, graphics, or tabular forms. The tool set is designed to access the devices available to the analyst without regard to the specific models or databases that have been used (11). Furthermore, the result data can be returned into the source database pool, along with the model database template as a source database template (12).

**OPERATIONAL LEVELS — MIMS BENEFITS**

Figure 6 presents the MIMS as a system consisting of six levels. From the top down, these levels are representative of the analyst, the analyst's modeling support system, the template/data interpretation library knowledge-based system, the template and Base Case intelligent databases, the database and model experts, and finally, the model data flow. In this subsection, MIMS benefits are identified with each of the levels.

The top, analyst's level is depicted in Fig. 6 by the dotted analyst box. Examining this level first, the MIMS provides direct, transparent access to the model data flow. Data manipulation interfaces and functions are uniform and consistent for any source database or model. The analyst responds to identical tools using standard protocols, and is therefore not required to become a technical expert. Tasks to define the Base Cases, scenarios and excursions, or to analyze model results are performed directly by the analyst without delay, miscommunication, or intervention by the support staff. Because of this, decisions can be made and questions answered within the same time frame as they are posed. The analyst has more direct control of the modeling process with fewer administrative responsibilities. Thus, the analyst can spend more time in understanding the appropriateness of the source data and the model. The overall benefits of the MIMS will be to yield a more comprehensive study with higher quality, greater flexibility, and improved cost effectiveness.

The analyst support system is shown in Fig. 6 by the three ellipses designated database integration tool, scenario generation tool, and results analysis tool. The most significant advantages of the analyst's modeling support system are the modularity with which it can be developed and the flexibility with which it can be used and maintained. Because these tools are not tied to particular databases or models, expanding the support system can be accomplished in an independent and incremental manner. New tools become immediately usable by the analyst, since they contain familiar interfaces and protocols. Off-the-shelf software, such as spread sheets or graphics packages, can be integrated as long as the standardized syntax is employed. A significant advantage is gained in the removal of errors from these tools. Because the same tools are used with various models and under tremendously different conditions, coding mistakes will be found and altered at a system level rather than in the context of a single model or study. Finally, the results analysis functions do not have to be specially written for each display device *and model.* Only the devices must be separately interfaced, but this, too, can be done at a system level.

The template/data interpretation library knowledge-based system is portrayed in Fig. 6 by the ellipse so entitled followed by three boxes labeled source databases, input databases, and input/output databases. This component of the MIMS provides a very open architecture. Like the modeling support system, the library can be developed modularly, incorporating standardized utilities or enhancements "as needed" without requiring model or analyst interface changes. A significant part of the library is the underlying database management system (DBMS). The DBMS can be selected from various commercial systems now or shortly available to reduce the MIMS development costs. Moreover, DBMS tool sets or "workbenches" are an increasingly greater part of the software integrated with the DBMS, which will further reduce MIMS development cost as well as maintaining compatibility with other sites using these systems.

The MIMS approach is complementary to the model integration concepts of knowledge-based system methodologies. Instead of requiring the model to perform in a particular environment, such as object-oriented or an executive structure,[4] the MIMS is designed to encompass *any* modeling framework. This is an important characteristic when dealing with the variety of military models developed throughout the services and their contractors. Unlike most knowledge-based methodologies which create a preferred future environment for modeling, the MIMS system addresses the issues of implementing current-generation models and simulations that are coded in Fortran, Simscript, or some other language and that do not inherently take advantage of the object-oriented paradigm.

The templates and Base Case intelligent databases are shown in Fig. 6 as dashed and solid-dashed boxes, respectively. They provide an explicit means for capturing the salient information associated with a source database, a model, or a study as a whole. The templates contain succinct and comprehensive documentation of the function and requirements of the model and its databases. Semantic information including assumptions and anomalies are captured. The template can automatically retain "lessons learned" from one project to the next, thus helping to prevent the recurrence of common misconceptions and errors. The templates themselves are held in a centralized repository that is widely

[4] Currently, two advanced multiple modeling environments that employ many desirable features are object-oriented and executive systems. An object-oriented system combines simulation and artificial intelligence techniques so that a model can be defined by a set of object behaviors and messages sent between objects (see, McArthur, et al., 1984). Executive systems provide an excellent environment for collecting algorithms and library functions within an appealing user environment (see, Carlson, et al., 1984). The disadvantage with both of these modeling approaches is the requirement to alter model code so that the software can fit underneath the environment's "umbrella."

accessible. In addition, tools can be built to simplify the initial generation as well as the updating of the template.

With the template and knowledge-based system in place, the database and model experts (depicted in the dotted boxes of Fig. 6) are free to perform other tasks. The mundane chores previously performed at this level are automated, allowing these experts to pursue more creative ventures including enhancing the MIMS support system, knowledge-based system, template mechanisms, or the models. Although these experts are initially required to set up the template, the overall long-term demand for manpower at this level is reduced and system efficiency is actually enhanced.

At the bottom, model data flow layer, the MIMS architecture requires *no alteration* of the model itself. Modifying source code is always risk-prone and costly, and maintaining the model in its original environment has many advantages. Synchronizing updates and versions with the proprietor or supplier of the model is also tedious and resource-intensive. By providing an integration framework with high-level analyst interfaces and no required model modifications, the analyst and the model operate in the most ideal environment for each. Furthermore, model developers no longer need to concern themselves with the awkward tasks of providing model support systems (such as data management, graphics, or display operations) within the context of their models. Since these functions are provided at a system level with only the requirement of a model template, the developer can concentrate on the unique computational aspects of the model.

## ANALYSIS FUNCTIONS — MIMS DEVELOPMENT TASKS

A third way to perceive the MIMS architecture presented in Fig. 6 is to consider the three general analyst functions — database integration, scenario generation, and results analysis. By partitioning the MIMS in this manner, the necessary development activities become more apparent.

Source database integration is currently the primary task under examination in the Intelligent Database Project. Although currently being conducted independently of the MIMS effort, the MOSF will coordinate closely with this project to insure synergistic results and to provide the means for technology transfer and implementation of their prototype system. This system will perform steps 1-3 as displayed in Fig. 6; that is, the processes of integrating, generalizing, and aggregating source databases into the format of the Base Case for the model. The Intelligent Database Project will develop methodologies for accomplishing this function using object-oriented and knowledge-based system techniques, although the MOSF will be responsible for the ultimate integration and implementation within the MIMS.

The scenario generation and results analysis functions are the research areas for the MOSF development team. Referring to Fig. 6, this constitutes steps 4-6 and 9-11, respectively. The four specific development tasks associated with the MOSF portion of the MIMS effort are summarized here and given in more detail by task below. First, a rigorous definition of the DBMS and the MSS data formats must be supplied. Second, the composition of the template must be defined, both for including transformation information from the model input/output database to the DBMS format and from the DBMS to the MSS format. Third, the Data/Template Interpretation Library, which actually performs these transformations, must also be specified. Finally, the MSS functions — constituting the scenario generation and results analysis tools — must be developed. These four research topics are discussed below as individual tasks, although the associated development effort may well be performed in parallel.

**Task 1: DBMS and MSS Data Format Definition**

The MIMS requires two distinct data formats. The first is used specifically to represent the Base Case in a DBMS. The DBMS format is preferred to the rigid and awkward structures inherent in source and model databases, and must conform to the management system selected as part of the data/template interpretation library (task 3). The format should provide the ability to untangle model database organization and structure. The format must be robust enough to handle every model in a uniform manner. The structure must be amenable to analytic manipulation, particularly the integration of databases into the Base Case as well as the extraction of a specific model database from the Base Case. Finally, since vast quantities of data could be included in a Base Case along with the desire to perform many operations on that data, the DBMS format must also be structured for efficiency and ease of use.

The second data format needed is the MSS format. Each tool developed for the analyst will require data in a MIMS MSS format. This format will include a data definition structure as well as communication protocols allowing tools to communicate. This structure will allow tools to be built independently but function harmoniously. The object-oriented paradigm used in advanced knowledge-based systems has been developed as a "natural" way to represent data. Rules of inheritance (i.e., gaining descriptive properties based on categorical subsetting like a "fighter" object possessing all of the characteristics of an "aircraft" object) have been well developed and are currently being implemented into operational systems. However, this paradigm lacks completeness in that various other kinds

of relationships are omitted (e.g., "ownership" in the sense that a fighter object can "own" a radio object). Furthermore, certain data types and structures (e.g., arrays and matrices used to represent weather or terrain information) are not well handled. The MIMS MSS will need to include a robust set of relationships and data types and be expandable as others are identified within specific applications.

**Task 2: Intelligent Database Template Definition**

Underlying the MIMS is the mechanism for capturing within a template a model's input and output data structure. The template information consists of a comprehensive and detailed description of all the model's parameters including their data types, relationships, acceptable values, defaults, missing data values, formats, descriptions, ownerships, hierarchies, and reference indices. The first step in designing the intelligent model database is to provide the syntactic and structural representation of a template. A methodology defining the information necessary for maintaining database consistency and functionality will be determined. Formalized syntax for the variety of data formats will be established as well as the manner in which relationships and hierarchies will be defined. Integration of a model or source database into the MIMS will require a thorough description within the template of data characteristics. The template must include all the information for transforming a model database to (and from) the DBMS format, and the Base Case to (and from) the MSS format.

Automating the template definition process will allow even faster, more accurate integration of new models into the MIMS. The construction of initial templates will be done almost entirely by hand. However, for other, large-scale models like TAC-SAGE, it will be necessary to provide an automated tool for defining the template interactively. A variety of tools are needed to simplify the extraction, organization, and integration of information held by technical experts or in reference guides. Indeed, these tools must be written with simple, visual interfaces so that the experts themselves will readily be able to create the templates.

**Task 3: Data/Template Interpretation Library**

A robust knowledge-based system employing intelligent databases will be created to bridge the gap between the analyst and the model. These routines will parse template information to read actual model data. Both input and output databases must be transformed into DBMS format using the information in the template and the functions in the library. Particular attention must be paid to databases already containing comments, "namelists," or other descriptive information. A fundamental part of the library is the database management

system that receives the restructured database from the model and provides the database, along with descriptive information from the template, to the model support system. The DBMS requirement may be satisfied by any number of relational or object management systems currently or soon to be made available. Interpretation library routines will also be created that can be embedded in each of the analyst tools to interpret the model database from MSS format. Of course the templates, too, must be maintained within a management system, and hence, a natural repository is within this same DBMS. This is the heart of the knowledge-based system that functionally replaces the model and database experts. All of these operations must be transparent to the analyst.

**Task 4: Scenario Generation and Results Analysis Tool**

A generalized set of analyst-oriented tools will be constructed as the model decision support system. These tools use the Base Case database and the model template, allowing the analyst to derive valid scenarios. The scenario tool will employ a visual user interface, permitting the analyst to easily browse and edit a scenario. Database operations (such as augmenting a range of parameters or increasing the values in a particular array) will be added to increase analyst productivity in assembling databases for sensitivity analysis or excursions. User help and referencing aids provide an efficient means for the novice as well as the more experienced analyst to quickly review the model database. Before execution of the model, the template information will be used to determine if the user has assembled a well-defined scenario, and indicate where errors have occurred.

Finally, the results analysis tools will also be created to assist the analyst in using and portraying conclusions from the modeling process. The tools for performing parametric analysis and experimental design are an essential part of the MIMS structure. A significant missing capability in current systems, but a required part of the MIMS, is robust case control functions to provide links between corresponding input and output scenarios and to determine differences. Graphics tools will be created to perform cartographic rendering and plotting. These tools must be designed to accommodate the analyst in portraying and examining model results in a variety of different ways and to provide access to a vast array of display and hard copy devices. Other tools must be implemented to create tables or to access statistical packages. Embedded in this task is the need to include various devices (Suns, Macs, PCs, printers, plotters, projectors, etc.) and the desirability of integrating off-the-shelf software that is already in common use. All of the scenario generation and results analysis functions will employ advanced "window" and menu-oriented terminal display facilities to enhance the usability of the MIMS.

## VI. CONCLUSIONS

The MIMS conceptual design is an initial step toward providing an automated support structure for improving quantitative analyses at RAND. The MIMS is the fundamental internal research effort within the Military Operations Simulation Facility. The first sections of this Note showed emphatically that current modeling architectures are severely deficient and that some improvements must be achieved. The latter sections describe the MIMS as a model support environment designed to satisfy the analyst's highest-priority needs. The MIMS employs state of the art methodologies including decision support systems, knowledge-based systems, and intelligent databases to relieve many of the critical and tedious tasks now facing analysts and project staffs. The fundamental concept is to reduce the need for model and database technical experts and replace them with a software system that provides modeling tools appealing to the analyst while leaving the model unaltered. The benefits of this conceptual architecture are numerous and can be summarized by the MOSF goal to enhance the quality of tactical and strategic military analysis and the effectiveness with which it is performed.

The MIMS is an ambitious, state of the art software creation that will require the joint efforts of a variety of information and computer specialists, and some experimentation. Many development challenges are generic to any major state of the art software project, such as the short-term tradeoff between resources and long-term productivity improvement. One particular MIMS resource requirement is that the integration of a source database or model have the rigorous specification of a template, and the time of various technical experts to accomplish it. There cannot be any error in description or the associated database will not be interpreted correctly. Although a restriction, the template forces the modeler to provide comprehensive documentation for source and model databases. Furthermore, assumptions, operational details, and anomalies are captured in an organized, accessible manner instead of existing only in the minds of technical experts or between the pages of voluminous manuals. In the long run, this reduces the demand for technical experts to perform these mundane tasks.

Another aspect to consider is that the MIMS will not be the fastest system. Although the model itself will not run any more slowly, the modeling support system will be less efficient than one designed for a specific purpose or model. Software generality always causes time delays. In response, consider the time delays caused now in finding the

technical experts and getting them involved in the modeling efforts. Some efficiency may be lost, but the overall analyst effectiveness will be dramatically increased. Furthermore, technological advances in hardware will ultimately reduce these system delays. System responsiveness must also be weighed against system robustness. Will the MIMS contain *enough* functionality to remove most of the tedious tasks in the modeling process? The MIMS architecture is specifically designed to reassure the modeler or the analyst that if the MIMS is not complete enough "today," the modular structure allows it to be easily expandable "tomorrow" so that modeling bottlenecks can be overcome.

We are also realistic about the potential pitfalls of this developmental activity. The MIMS will not solve all modeling problems. Although this and other documentation clearly describe the functions MIMS is meant to solve, many will perceive that it should accomplish much more or something completely different. It is also unlikely that within the context of the problems it does solve that it will always work. Although we can be conceptually very optimistic, this system will break, *but not often and not for long time durations*. The system architecture allows for rapid repair. Finally, it is inevitable that the MIMS will cost more and take longer to develop than desired. These are realities. The steady reliance on "old" technologies to assist on-going modeling efforts is a strong indicator of how difficult this problem and its ultimate comprehensive resolution are. In defense of the MIMS, let us ask: Are analysts satisfied with the way quantitative analysis must be performed now? Do the current systems or environments solve the problems commonly faced in the modeling process? Are the current manpower-intensive modeling efforts free of flaws? Does it currently cost less or require less time than desired to apply a typical model? Ultimately, the question must be answered whether the MIMS with all its development requirements is preferable to continuing with the same architectures, mechanisms, and practices currently available.

The MIMS concept could well revolutionize the way modeling is currently being tediously performed. The potential MIMS conceptual, design, and operational benefits are immense. Many years from now, perhaps the MIMS will not be the ultimate modeling system in use, but the concept will undoubtedly be one of the fundamental premises behind that inevitable future system.

## BIBLIOGRAPHY

Cammarata, S., "An Intelligent Information Dictionary for Semantic Manipulation of Relational Databases," in J. W. Schmidt, S. Seri, and M. Missikoff (eds.), *Lecture Notes in Computer Science #303*, Proceedings of the International Conference on Extending Database Technology, March 1988.

Carlson, P., C. Emmanuelli, E. Harris, and D. Perkins, *Primer for the Transportable Applications Executive*, NASA Goddard Flight Center, Maryland, January 1984.

Dazzo, G., "How to Maintain Large Scientific Programs," *Computers in Physics*, January/February, 1988.

Donohue, G. L., B. E. Bennett, and J. B. Hertzog, *The RAND Military Operations Simulation Facility: An Overview*, The RAND Corporation, N-2428-RC, April 1986.

Hayes-Roth, F., D. A. Waterman, and D. B. Lenat (eds.), *Building Expert Systems*, The Teknowledge Series in Knowledge Engineering, Addison-Wesley, Reading, Massachusetts, 1983.

McArthur, D., P. Klahr, and S. Narain, *ROSS: An Object-Oriented Language for Constructing Simulations*, The RAND Corporation, R-3160-AF, December 1984.

Sprague, R. H., Jr.,and E. D. Carlson, *Building Effective Decision Support Systems*, Prentice-Hall, Inc., New Jersey, 1982.